

1. ***Trình bày sự phát triển của điện thoại di động:***

**Lịch sự hình thành điện thoại:**

Ngày mùng 10 tháng 3 năm 1876 được coi là mốc son đầu tiên đánh dấu sự ra đời của điện thoại mà người cha phát minh ra nó là Alexander Graham Bell. Đây thực sự là một bước tiến công nghệ đột phá, nó đã mở ra cả một kỷ nguyên phát triển mới trong lịch sử thông tin liên lạc, thay thế cho phương thức cũ thô sơ là **điện báo** trước đó.

Sau đó, vào tháng 6 năm 1876, máy điện thoại lần đầu tiên được đưa vào sử dụng trong hội chợ triển lãm *“Contennial Exposition”* ở Philadelphia. Trong giai đoạn đầu, chiếc điện thoại được thiết kế khá kiểu cách và cầu kỳ với hai nét đặc trưng là có hai đầu, một ống để nói và một ống để nghe và chỉ có người giàu mới có khả năng sở hữu chúng.

Tiếp đến là sự phát triển một số hình dáng khác của điện thoại như bốt điện thoại (nhưng thật tiếc vì giờ nó chỉ còn mang tính biểu tượng trên phim ảnh), điện thoại trong xe (đây là một mẫu điện thoại tiến gần với chức năng "di động", nó nhỏ gọn và có thể gắn vào trong xe hơi, rất tiện lợi cho người sử dụng thời bấy giờ)

Ra mắt vào năm 1967, "Carry phone" được coi là chiếc điện thoại "di động" đầu tiên, là một bước tiền gần hơn tới mẫu điện thoại di động nguyên bản. Tuy nhiên, dù gọi là di động nhưng người dùng lúc nào cũng phải vác cái hộp máy to lù lù và nặng đến 4-5 kg như cái va-li, trong khi giá thành lại không hề rẻ làm cho sản phẩm này rốt cuộc cũng không được phổ biến rộng rãi.

**Lịch sử hình thành điện thoại di động:**

Ra mắt vào năm 1967, "Carry phone" được coi là chiếc điện thoại "di động" đầu tiên, là một bước tiền gần hơn tới mẫu điện thoại di động nguyên bản. Tuy nhiên, dù gọi là di động nhưng người dùng lúc nào cũng phải vác cái hộp máy to lù lù và nặng đến 4-5 kg như cái va-li, trong khi giá thành lại không hề rẻ làm cho sản phẩm này rốt cuộc cũng không được phổ biến rộng rãi.

Và cái ngày quan trọng đánh dấu một sự bứt phá trong lịch sử điện thoại thế giới đã đến, ngày mùng 03 tháng 04 năm 1973, cái ngày mà tiến sỹ Martin Cooper của Motorola thực hiện cuộc gọi đầu tiên từ chiếc điện thoại di động của mình. Vào thời điểm đó, không một ai đi dạo trên đại lộ Sixth Anvenue của thành phố New York (Mỹ) lại có thể nhận ra cái khối thô kệch, to đùng có hình dáng chính xác của một viên gạch với trọng lượng gần 1kg trong tay Martin là tiền thân của những “con dế” hiện đại ngày nay. Sau này, thiết bị này được công nhận là chiếc điện thoại di động đầu tiên trên thế giới, được gọi với cái tên là Motorola Dyna TAC và dĩ nhiên Martin Cooper được xem là cha đẻ của điện thoại di động. Chi phí để tạo ra Motorola Dyna TAC lúc bấy giờ vào khoảng 3,500 USD và được tung ra thị trường chính thức 10 năm sau cuộc gọi lịch sử đầu tiên với giá bán lẻ lên đến gần 4,000 USD.

**Sự phát triển điện thoại di động:**

Từ đó đến nay trải qua hơn 40 năm tồn tại và phát triển, điện thoại di động ngày càng trở nên phổ biến và không thể thiếu được trong cuộc sống con người. Chúng không ngừng được thay đổi, cải tiến trong công nghệ cũng như kiểu dáng. Các thương hiệu sản xuất di động hàng đầu trên thị trường như Nokia, Blackberry, Samsung, LG, Sony Ericsson, Motorola…

Đặc biệt là vào năm 2007, hãng Apple đã “trình làng” chiếc điện thoại Iphone, sự ra đời này đánh dấu sự sáng tạo đột phá về kiểu dáng với nét đặc trưng là màn hình cảm ứng siêu nhạy giúp người sử dụng có thể thao tác dễ dàng chỉ bằng việc “lướt” nhẹ các ngón tay. Iphone đã tạo nên một cơn sốt chưa từng có và chính thức khởi đầu cho cuộc chạy đua cạnh tranh khốc liệt trong phân khúc dòng điện thoại thông minh (smartphone).

Như vậy, chắc chắn trong tương lai không xa, cuộc đua giữa các hãng điện thoại di dộng sẽ ngày càng gay gắt và không thể lường trước được vì họ luôn tích cực tìm tòi, đổi mới, cải tiến để cho ra đời các mẫu điện thoại, mà cụ thể hiện tại là smartphone với những hình dáng độc lạ (như dẻo có thể uốn cong hoặc gập lại thành nhiều phần dự đoán sẽ ra mắt thị trường vào năm 2017) và các tính năng ngày càng được hoàn thiện và bổ sung nhằm đem lại cho người dùng sự tiện lợi tối đa với phương châm: cả thế giới thu nhỏ trong lòng bàn tay.

**Các trường hợp thành công và thất bại:**

**Thất bại: Nokia**

**Sự thất bại nổi tiếng của Nokia trong thị trường điện thoại bắt đầu từ những ngày tháng ngồi trên đỉnh cao của hãng. Quản lý yếu kém, định hướng sai lầm và đấu đá nội bộ là những nguyên nhân khiến Nokia tự làm yếu chính mình.**

Trong nhiều năm qua, Nokia là một biểu tượng trong ngành công nghiệp di động. Chỉ trong vòng hai thập kỷ, công ty Phần Lan này đã tạo ra và chi phối cả ngành công nghiệp di động toàn cầu với 40% thị phần ở thời kỳ đỉnh cao.

Nhiều nhà phân tích cho rằng, sự sụp đổ của Nokia là do sự đi lên mạnh mẽ của những hãng công nghệ khác như Apple, Samsung và Google. Tuy nhiên, sự thật lại không đơn giản như vậy. Nếu các công ty khác mạnh lên, tại sao Nokia lại không thể mạnh lên? Nokia thật sự đã tự làm yếu chính mình khi rơi vào một cuộc khủng hoảng nội bộ trước khi bị các đối thủ đe dọa trên thị trường và làm trầm trọng thêm các vấn đề tài chính của hãng. Trên thực tế, những mầm mống của sự sụp đổ đã xuất hiện khi Nokia đang ngồi trên đỉnh cao thành công, theo bài phân tích của báo South China Morning Post.

**Thành Công: Iphone (Apple)**

1. Đối với bất cứ sản phẩm nào Apple sáng tạo, người sáng tạo ra sản phẩm đó phải chính là người cũng mong muốn sản phẩm này.

## Sản phẩm phải dễ dàng sử dụng

## Duy trì mọi thứ đơn giản

## Dịch vụ khách hàng và trải nghiệm tại cửa hàng tốt nhất

## Apple chỉ làm một sản phẩm nếu Apple có thể làm điều đó tốt hơn

## Apple đi trước đối thủ 2 năm, Luôn là người đi tiên phong, dẫn dắt thị trường

## Tạo thương hiệu tốt, iphone ko chỉ là điện thoại mà còn là vật trang sức của những người có tiền.

## Hệ thống phần mềm độc đáo, hệ điều hành tốt.

1. ***Lý do và sự phát triển của điện thoại di động, phân biệt điện thoại di động, điện thoại thông minh, thiết bị di động***

Sự phát triển: giống câu 1

Lý do phát triển

# 10 khác biệt của điện thoại cục gạch với smartphone

1. **Thời lượng pin hàng tuần**
2. **Mỗi thiết bị có đặc trưng riêng**
3. **Không có Internet**
4. **Game di động: Rắn săn mồi – smart phone có nhiều hơn**
5. **Màn hình cảm ứng: Cơn ác mộng của người dùng**
6. **Windows là tương lai của ngành công nghiệp di động**
7. Người dùng trả tiền để... thay thế logo nhà mạng

Trước khi có smartphone, mọi thứ người dùng muốn sở hữu từ nhạc chuông, hình nền, nhắn tin, GPRS, thay logo mạng,... đều phải mua từ những nhà cung cấp dịch vụ. Nhưng với sự phổ biến của các dịch vụ nhắn tin OTT và Wi-Fi đã khiến thị trường di động thay đổi

1. **Crazy Frog - Tượng đài nhạc số - các phần mềm nhạc**
2. **Điện thoại "nồi đồng cối đá"**

Nếu bạn đánh rơi chiếc Nokia xuống sàn, sẽ có thứ bị nứt nhưng không phải là điện thoại. Nếu bạn đặt điện thoại vào đường ray, đoàn tàu sẽ bị trật bánh. Và nếu smartphone "cứng cáp" như những chiếc Nokia, chúng ta sẽ không bao giờ phải xem những thử nghiệm độ bền nhàm chán.

1. **chiếc di động Nokia gắn với ký ức người dùng Việt**

 Nokia từ lâu được xem là tượng đài khi nhắc tới điện thoại di động ở Việt Nam. Nhiều thế hệ người dùng gắn bó với các sản phẩm hãng này.

1. ***Uư điểm và khuyết điểm khi sử dụng máy ảo so với máy thật***

### Kiểm thử trên thiết bị thật

Kiểm thử trên thiết bị thật cho phép chúng ta chạy các ứng dụng di động của mình và kiểm thử chức năng của nó. Việc thực hiện kiểm thử trên thiết bị thật đảm bảo với bạn rằng ứng dụng của bạn sẽ hoạt động trơn tru trên các thiết bị của khách hàng. Bởi vì khi không được test trên một số thiết bị hạn chế, thì việc test trên device thật của bạn cũng sẽ nhận được một ứng dụng tốt hơn khi không được kiểm tra trên bất cứ một thiết bị thật nào.

#### Máy ảo ( Emulator và Simulator)

| **Simulator** | **Emulator** |
| --- | --- |
| Mục tiêu của simulator là mô phỏng trạng thái bên trong của một đối tượng. | Mục tiêu của emulator là mô phỏng hoặc bắt chước hành vi bên ngoài của đối tượng càng gần càng tốt. |
| Simulator thích hợp khi cần kiểm tra hành vi bên ngoài của thiết bị di động như tính toán, thực hiện giao dịch và .... | Emulator thích hợp khi cần kiểm tra hành vi nội bộ của thiết bị di động như phần cứng, phần cứng nội bộ của nó và .... |
| Simulator được viết bằng ngôn ngữ cao cấp. | Emulator được viết bằng các ngôn ngữ máy. |
| Simulator có thể gây ra khó khăn trong việc debug | Emulator thì lại phù hợp hơn trong việc debug |
| Simulator chỉ là một phần được xây dựng lại từ phần mềm gốc | Emulator là một sự tái triển khai hoàn chỉnh của phần mềm gốc. |

### Ưu/ nhược điểm giữa việc sử dụng thiết bị thật và thiết bị giả lập/ mô phỏng

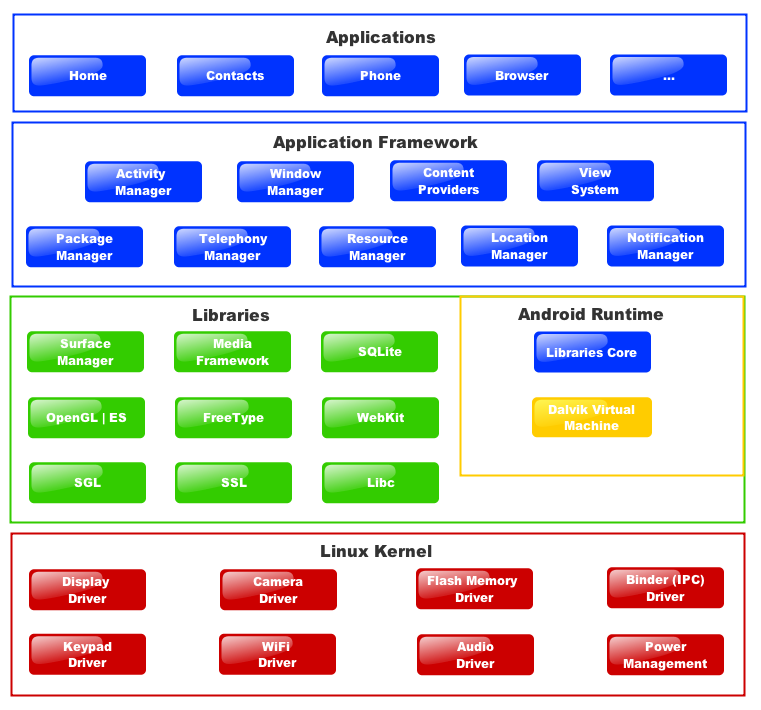
#### 1. Ưu điểm của thiết bị thật, Simulator và Emulator trong việc kiểm thử.

| **Vấn đề** | **Kiểm thử trên máy ảo** | **Kiểm thử trên thiết bị thật** |
| --- | --- | --- |
| Ứng dụng dựa trên tình huống | Có những tình huống cụ thể mà việc mua thiết bị để đáp ứng như cầu không thể đáp ứng như : kinh tế , phiên bản , số lượng ... Vì vậy có thể cần phải sử dụng trình giả lập/mô phỏng trong các trường hợp như thế này. | Thiết bị thật cho phép người kiểm thử kiểm tra gần như tất cả các tình huống thời gian thật có thể được kiểm tra cho các ứng dụng di động. Các thiết bị này được mô phỏng cách sử dụng thực tế. Nó cũng hỗ trợ được trong hoàn cảnh thực tế như: có dễ sử dụng ứng dụng khi trên tàu hoặc khi đi bộ không? Có sử dụng được bên ngoài ánh sáng mặt trời hay khi mưa không? ... |
| Cảm giác dễ tương tác với những thiết bị thật | Các giao diện của các thiết bị di động tạo ra các vấn đề, theo đó những người kiểm thử không biết chính xác nên mua thiết bị thực nào để test và cần xem xét các ràng buộc về ngân sách. Trình giả lập/mô phỏng được thiết kế riêng cho loại tình huống này. | Thiết bị thật cho phép người kiểm thử kiểm tra ngay cả các vấn đề khả năng sử dụng như giao diện của ứng dụng, độ phân giải màu của màn hình, cho dù hình ảnh sáng hay không trong cả điều kiện ban ngày lẫn ban đêm ... |
| Tính năng sẵn có | Trình giả lập/mô phỏng trong hầu hết các trường hợp là phần mềm mở và miễn phí, có thể tải xuống rất dễ dàng từ Internet và sẵn sàng để được kiểm tra. | Các thiết bị thật cho phép các vấn đề kiểm tra hiệu năng nghiêm ngặt như làm việc với ứng dụng truyền tải thời gian thực trong 15 giờ liên tục mà không thể mô phỏng được khi sử dụng các trình giả lập. |
| Dễ mở một ứng dụng web thông qua URL | Việc kiểm thử một ứng dụng web sẽ trở nên dễ dàng hơn. Người dùng chỉ cần sao chép và dán URL ứng dụng. | Việc thực hiện kiểm thử trên các thiết bị thật sẽ có độ tin cậy cao hơn. |
| Chụp ảnh màn hình của các tình huống có lỗi xuất hiện | Chụp ảnh màn hình trên trình mô phỏng rất dễ dàng vì chỉ cần sử dụng các tiện ích văn phòng của Microsoft. | Việc test với các thiết bị thật là rất hữu ích về test khả năng tương tác. |
| Mô phỏng chính xác vấn đề về pin | Trình giả lập/mô phỏng không thể mô phỏng các vấn đề về pin. | Các thiết bị thật có thể dễ dàng thực hiện các tình huống này. |
| Xác nhận các tác động cắt ngang tương tác | Trình giả lập/mô phỏng không thể mô phỏng các tương tác ngắt như SMS đến, những cuộc gọi đến... | Các thiết bị thật có thể dễ dàng mô phỏng các trường hợp đó . |
| Xác nhận hiển thị màu chính xác | Trình mô phỏng/giả lập không thể mô phỏng đúng ,mô phỏng hiển thị màu chính xác của thiết bị khi thiết bị thật ở trong ánh sáng mặt trời hoặc màu đen. | Các thiết bị thực có thể dễ dàng mô phỏng các màn hình màu chính xác. |
| Xác nhận hiệu suất | Hiệu suất của trình giả lập có xu hướng chậm hơn so với các thiết bị thực | Các thiết bị thật có xu hướng hoạt động nhanh hơn so với trình giả lập hoặc trình mô phỏng. |

Tổng kết:

Emulators và Simulators với những ưu điểm là dễ tương tác , chi phí rẻ , dễ cài đặt .... Tuy nhiên nó chỉ giống với device thật ở mức tương đối, nên sẽ có những bug chạy trên emulator thì không bị bug nhưng khi test thật trên device thì lại bị bug . Và thêm nữa có nhiều môi trường test mà Emulators và Simulators bị giới hạn hơn so với thiết bị thật . Do đó việc kiểm thử sử dụng thiết bị giả lập/ mô phỏng nên được sử dụng trong các giai đoạn ban đầu của dự án và kết hợp cùng với thiết bị thật ở giai đoạn cuối của dự án nhằm đảm bảo chất lượng tổng thể của phần mềm.

1. ***Kiến trúc Android***



### ****1. Nhân Linux (Linux kernel)****

Nằm ở tầng dưới cùng là lớp Linux kernal – Linux 3.6 với khoảng 115 patches. Lớp này liên hệ với phần cứng và nó chứa tất cả driver phần cứng cần thiết như camera, bàn phím, màn hình…

### ****2. Thư viện Android (Android libraries) và Android Runtime****

Phía trên tầng Linux kernel là tầng libraries, chứa những thư viện hỗ trợ. Một số có thể kể đến như là bộ máy trình duyệt web mã nguồn mở WebKit, thư viện libc, cơ sở dữ liệu SQLite tiện lợi cho việc lưu trữ và chia sẻ dữ liệu, thư viện hỗ trợ thu phát âm thanh và video, thư viện SSL giúp bảo mật mạng…

Tầng này chứa tất cả các thư viện Java, được viết đặc biệt cho Android như các thư viện framework, các thư viện xây dựng giao diện, đồ họa và cơ sở dữ liệu. Dưới đây là một số thư viện quan trọng mà các lập trình viên nên biết:

* **android.app** − Cung cấp quyền truy cập tới mô hình ứng dụng và là nền tảng của mọi ứng dụng Android applications.
* **android.content** − Cho phép truy cập nội dung, phát hành và tin nhắn giữa các ứng dụng và các thành phần trong một ứng dung.
* **android.database** −  Được dùng để truy cập vào dữ liệu được đưa ra bởi bộ phận cung cấp nội dung, bao gồm các lớp quản lý cơ sở dữ liệu SQLite.
* **android.opengl** − Cho phép tương tác với thư viện đồ họa OpenGL ES 3D.
* **android.os** − Giúp ứng dụng truy cập những dịch vụ cơ bản của hệ điều hành bao gồm tin nhắn, dịch vụ hệ thống và liên lạc nội bộ (inter-process communication).
* **android.text** − Được dùng để vẽ và thao tác văn bản trên màn hình.
* **android.view** − Các khối xây dựng cơ bản của giao diện người dùng.
* **android.widget**
* **android.webkit**

Trong tầng này còn có một phần không kém phần quan trọng là Android runtime, bây giờ chúng ta sẽ tìm hiểu các thư viện C/C++ trong tầng này.

Android Runtime chứa Dalvik Virtual Machine (DVM) – một biến thể của Java Virtual Machine, đặc biêt thiết kế và tối ưu hóa cho Android. DVM giúp mỗi ứng dụng Android chạy trong chính tiến trình (process) của nó với một đại diện (instance) của DVM.

Ngoài ra, Android Runtime cũng chứa tập hợp các thư viện quan trong cho phép người lập trình viết ứng dụng sử dụng ngôn ngữ lập trình Java.

### ****3. Application Framework****

Application Framework cung cấp nhiều dịch vụ cấp cao dưới dạng các lớp viết bằng Java (Java classes) Lập trình viên được phép sử dụng các lớp này để tạo ra các ứng dụng.

Android framework chứa các dịch vụ quan trọng như:

* **Activity Manager** − Quản lý tất cả các phần của vòng đời (lifecycle) ứng dụng và các hoạt động ngăn xếp (activity stack).
* **Content Providers** − Cho phép ứng dụng phát hành và chia sẻ dữ liệu với ứng dụng khác.
* **Resource Manager** − Cho phép truy cập tới những tài nguyên không phải là mã nguồn như chuỗi, cài đặt màu, bố cục giao diện.
* **Notifications Manager** − Giúp ứng dụng hiển thị thông báo và nhắc nhở người dùng.
* **View System** − Một tập hợp mở rộng giúp tạo giao diện người dùng.

### ****4 – Ứng dụng (applications)****

Ở tầng trên cùng là các ứng dụng Android đi kèm với hệ điều hành như Contacts Books, Browser, Games… Thiết nghĩ không cần nói nhiều thêm về tầng này.

1. ***ThreadingNoThreading (trình bày vấn đề, giải pháp, code minh họa)***

Khi 1 thành 1 của app khởi động mà không có bất kì thành phần khác của ứng đó đang hoạt động, thì hệ thống android sẽ bắt đầu 1 tiến trình(process) Linux cho app đó với 1 luồng(thread).

Mặc định thì mọi thành phần của 1 app đều sử dụng chung 1 tiến trình và luồng(main thread).

Nếu 1 thành phần của app khởi động mà đã có sẵn 1 tiến trình của app tồn tại, thì nó sẽ dùng chung luồng có sẵn của app.

Ta có thể cho 1 thành phần của app chạy trên 1 tiến trình khác và tạo thêm luồng cho tiến trình.

Khi mà main thread thực hiện 1 công việc dài như lấy data từ mạng hay database thì nó sẽ chặn toàn bộ UI. Khi thread bị chặn thì không có 1 sự kiện nào khác có thể được thực hiện. dẫn đến app bị treo trong 1 thời gian dài.

Ta có thể tạo ra 1 thread mới để thực hiện các công việc tốn nhiều thời gian, nhưng chỉ có main thread mới có thể update UI.

# boolean View.post (Runnable action)

public void onClick(View v) {  
    new Thread(new Runnable() {  
        public void run() {  
            // a potentially time consuming task  
            final Bitmap bitmap =  
                    processBitMap("image.png");  
            imageView.post(new Runnable() {  
                public void run() {  
                    imageView.setImageBitmap(bitmap);  
                }  
            });  
        }  
    }).start();  
}

# void Activity.runOnUiThread (Runnable action)

public void onClick(View v) {

new Thread() {

public void run() {

try {

runOnUiThread(new Runnable() {

public void run() {

textView.setText("test");

}

});

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}.start();

}